Journal of Artificial Intelligence Research 54 (2015) 83-122 Submitted 02/15; published 09/15

Word vs. Class-Based Word Sense Disambiguation

Rubén Izquierdo RUBEN.IZQUIERDOBEVIA@VU.NL
VU University of Amsterdam
Amsterdam. The Netherlands

Armando Suárez ARMANDO@DLSI.UA.ES
University of Alicante
Alicante. Spain

German Rigau GERMAN.RIGAU@EHU.ES
University of the Basque Country
San Sebastián. Spain

Abstract

As empirically demonstrated by the Word Sense Disambiguation (WSD) tasks of the last SensEval/SemEval exercises, assigning the appropriate meaning to words in context has resisted all attempts to be successfully addressed. Many authors argue that one possible reason could be the use of inappropriate sets of word meanings. In particular, WordNet has been used as a *de-facto* standard repository of word meanings in most of these tasks. Thus, instead of using the word senses defined in WordNet, some approaches have derived semantic classes representing groups of word senses. However, the meanings represented by WordNet have been only used for WSD at a very fine-grained sense level or at a very coarse-grained semantic class level (also called SuperSenses). We suspect that an appropriate level of abstraction could be on between both levels. The contributions of this paper are manifold. First, we propose a simple method to automatically derive semantic classes at intermediate levels of abstraction covering all nominal and verbal WordNet meanings. Second, we empirically demonstrate that our automatically derived semantic classes outperform classical approaches based on word senses and more coarse-grained sense groupings. Third, we also demonstrate that our supervised WSD system benefits from using these new semantic classes as additional semantic features while reducing the amount of training examples. Finally, we also demonstrate the robustness of our supervised semantic class-based WSD system when tested on out of domain corpus.

1. Introduction

Word Sense Disambiguation (WSD) is an intermediate Natural Language Processing (NLP) task that consists in assigning the correct lexical interpretation to ambiguous words depending on the surrounding context (Agirre & Edmonds, 2007; Navigli, 2009). One of the most successful approaches in the last years is the *supervised learning from examples*, in which Machine Learning classification models are induced from semantically annotated corpora (Màrquez, Escudero, Martínez, & Rigau, 2006). Quite often, machine learning systems have obtained better results than the knowledge-based ones, as shown by experimental work and international evaluation exercises such as Senseval or SemEval[1]. Nevertheless, lately some weakly supervised or knowledge–based approaches are reaching a performance close to the supervised techniques on some specific tasks. In all these tasks, the

1. All the information about these competitions can be found at `http://www.senseval.org`.

corpora are usually manually annotated by experts with word senses taken from a particular lexical semantic resource, most commonly WordNet (Fellbaum, 1998).

However, WordNet has been widely criticized for being a sense repository that often provides too fine–grained sense distinctions for higher level applications like Machine Translation (MT) or Question & Answering (AQ). In fact, WSD at this low level of semantic granularity has resisted all attempts of inferring robust broad-coverage models. It seems that many word–sense distinctions are too subtle to be captured by automatic systems with the current small volumes of word–sense annotated examples. Using WordNet as a sense repository, the organizers of the English all-words task at SensEval-3 reported an inter-annotation agreement of 72.5% (Snyder & Palmer, 2004). Interestingly, this result is difficult to outperform by state-of-the-art sense-based WSD systems.

Moreover, supervised sense–based approaches are too biased towards the most frequent sense or the predominant sense on the training data. Therefore, the performance of supervised sense–based systems is strongly punished when applied to domain specific texts where the sense distribution differs considerably with respect to the sense distribution in the training corpora (Escudero, Màrquez, & Rigau., 2000).

In this paper we try to overcome these problems by facing the task of WSD from a Semantic Class point of view instead of the traditional word sense based approach. A semantic class can be seen as an abstract concept that groups sub–concepts and word senses sharing some semantic properties or features. Examples of semantic classes are VEHICLE, FOOD or ANIMAL. Our hypothesis is that using an appropriate set of semantic classes instead of word-senses could help WSD in several aspects:

- A higher level of abstraction could ease the integration of WSD systems into other higher level NLP applications such as Machine Translation or Question & Answering
- Grouping together semantically coherent sets of training examples could also increase the robustness of supervised WSD systems
- The so–called bottleneck acquisition problem could also be alleviated

These points will be further explained along the paper. Following this hypothesis we propose to create classifiers based on semantic classes instead of word sense experts. One semantic classifier will be trained for each semantic class and the final system will assign the proper semantic class to each ambiguous word (instead of the sense as in traditional approaches). For example, using our automatically derived semantic classes (that will be introduced later), the three senses of *church* in WordNet 1.6 are subsumed by the semantic classes RELIGIOUSORGANIZATION, BUILDING and RELIGIOUSCEREMONY. Also note that these semantic classes still discriminate among the three different senses of the word *church*. For instance, if we assign the semantic class BUILDING to an occurrence of *church* in a context, we still know that it refers to its second sense. Additionally, the semantic class BUILDING now covers more than six times more training examples than those covered by the second sense of *church*.

An example of text from senseval–2 automatically annotated with semantic classes can be seen in Figure 1. It shows the automatic annotations by our class–based classifiers with different semantic classes. **BLC** stands for Basic Level Concepts[2] (Izquierdo, Suarez, & Rigau, 2007), **SS**

2. We will use the following format throughout this paper to refer to a particular sense: word^{num}_{pos}, where *pos* is the part-of-speech: n for nouns, v for verbs, a for adjectives and r for adverbs, and *num* stands for the sense number.

for SuperSenses (Ciaramita & Johnson, 2003), **WND** for WordNet Domains (Magnini & Cavaglià, 2000; L. Bentivogli & Pianta, 2004) and **SUMO** for Suggested Upper Merged Ontology (Niles & Pease, 2001). Incorrect assignments are marked in italics. The correct tags are included between brackets next to the automatic ones. Obviously, these semantic resources relate senses at different level of abstraction using diverse semantic criteria and properties that could be of interest for subsequent semantic processing. Moreover, their combination could improve the overall results since they offer different semantic perspectives of the text.

Id	Word	BLC	SS	WND	SUMO
1	An				
2	ancient				
3	stone	artifact^1_n	noun.artifact	building	Mineral
4	church	building^1_n	noun.artifact	building	Building
6	amid				
7	the				
8	fields	*geographic_area*^1_n [physical_object^1_n]	*noun.location* [noun.object]	*factotum* [geography]	LandArea
9	,				
10	the				
11	sound	property^2_n	noun.attribute	*factotum* [acoustics]	*RadiatingSound* [SoundAttribute]
12	of				
13	bells	device^1_n	noun.artifact	*factotum* [acoustics]	MusicalInstrument
14	cascading	move^2_v	verb.motion	factotum	Motion
15	from				
16	its				
17	tower	construction^3_n	noun.artifact	factotum	Building
18	calling	*designate*^2_v [request^2_v]	*verb.stative* [verb.communication]	factotum	*Communication* [Requesting]
19	the				
20	faithful	*group*^1_n [social_group^1_n]	noun.group	*person* [religion]	Group
21	to				
22	evensong	*time_of_day*^1_n [writing^2_n]	noun.communication	religion	*TimeInterval* [Text]

Table 1: Example of the automatic annotation of a text with several semantic class labels

The main goal of our research is to investigate the performance of alternative *Semantic Classes* derived from WordNet on supervised WSD. First, we propose a system to automatically extract sets of semantically coherent groupings from nominal and verbal senses from WordNet. The system allows to generate arbitrary sets of semantic classes at distinct levels of abstraction. Second, we also analyze its impact with respect to alternative *Semantic Classes* when performing class–based WSD. Our empirical results show that our automatically generated classes performs better than those created manually (WNDomains, SUMO, SuperSenses, etc.) while capturing more precise information. Third, we also demonstrate that our supervised WSD system benefits from using these new semantic classes as additional semantic features while reducing the amount of training

examples. Finally, we show that our supervised class-based system can be adapted to a particular domain. Traditional word sense based systems are also included only for comparison purposes.

Summarizing, our research empirically investigates:

- The performance of alternative semantic groupings when used in a supervised class-based WSD system
- The impact of class-based semantic features in our supervised WSD framework
- The required amount of training examples needed by a class-based WSD in order to obtain competitive results
- The relative performance of the class-based WSD systems with respect WSD based on word experts
- The robustness of our class-based WSD system on specific domains

Moreover, when tested on out of domain dataset, our supervised class-based WSD system obtains slightly better results than a state-of-the-art word sense based WSD system, the ItMakesSense system presented by Zhong and Ng (2010).

After this introduction, we present the work directly related with our research on supervised WSD based on semantic classes. Then, Section 3 presents the sense-groupings and semantic classes used in this study. Section 4 explains our method to automatically derive semantic classes from WordNet at different levels of abstraction. Moreover an analysis of different semantic groupings is included. Section 5, presents the system that we have developed to perform supervised class-based WSD. The performance of this system is shown in Section 6, where the system is tested on several WSD datasets provided by international evaluations. A comparison with other participants on these competitions is introduced in sections 7 and 8. Some experiments with our system applied to a specific domain are analyzed in Section 9. Finally, some conclusions and future work are presented in section 10.

2. Related Work

The field of WSD is very broad. There have been a large amount of publications about WSD over the last 50 years. This section only revises some relevant WSD approaches dealing with the appropriate sets of meanings a word should have.

Some research has been focused on deriving different word-sense groupings to overcome the fine–grained distinctions of WordNet (Hearst & Schütze, 1993; Peters, Peters, & Vossen, 1998; Mihalcea & Moldovan, 2001; Agirre & de Lacalle, 2003; Navigli, 2006; Snow, S., D., & A., 2007). That is, they provide methods for grouping senses of the same word, thus producing coarser word sense groupings. For example, for the word *church* having three senses in WordNet 1.6, the sense grouping presented by Snow et al. (2007) only produces a unique grouping. That is, according to this approach *church* is monosemous.

In the OntoNotes project (Hovy, Marcus, Palmer, Ramshaw, & Weischedel, 2006), the different meanings of a word are considered as a kind of tree, ranging from coarse concepts on the root to fine–grained meanings on the leaves. The merging was increased from fine to coarse grained until obtaining an inter annotator agreement of around 90%. This coarse-grained repository was used

in the WSD lexical sample task of SemEval-2007 (Pradhan, Dligach, & Palmer, 2007), where the systems scored up to 88.7% F–score. Note that this merging was created for each word following a manual and very costly process.

Similarly to the previous approach, another task was organized within SemEval-2007 which consisted in the traditional WSD all word task using another coarse–grained sense repository derived from WordNet (Navigli, Litkowski, & Hargraves, 2007). In this case all the WordNet synsets were automatically linked to the Oxford Dictionary of English (ODE) using a graph algorithm. All the meanings of a word linked to the same ODE entry were merged into a coarse sense. The systems achieving the top scores followed supervised approaches taking advantage of different corpora for the training, reaching a top F–score of 82.50%.

Both of the previous cases are aimed at solving the granularity problem of the word sense definitions in WordNet. However, both approaches are still word experts (one classifier is trained for each word). Obviously, decreasing the average polysemy of a word by using coarser–senses makes easier the classification choice. As a result, the performance of these systems increase at the cost of reducing its discriminative power.

Conversely, instead of word experts, our approach creates semantic class experts. Each of these semantic classifiers can exploit diverse information extracted from all the meanings *from different words* that belong to that class.

Wikipedia (Wikipedia, 2015) has been also recently used to overcome some problems of the supervised learning methods: excessively fine–grained definition of meanings, lack of annotated data and strong domain dependence of the existing annotated corpora. In this way, Wikipedia provides a new source of annotated data, very large and constantly in expansion (Mihalcea, 2007; Gangemi, Nuzzolese, Presutti, Draicchio, Musetti, & Ciancarini, 2012).

In contrast, some research has been focused on using predefined sets of sense-groupings for learning class–based classifiers for WSD (Segond, Schiller, Greffenstette, & Chanod, 1997; Ciaramita & Johnson, 2003; Villarejo, Màrquez, & Rigau, 2005; Curran, 2005; Ciaramita & Altun, 2006; Izquierdo, Suárez, & Rigau, 2009). That is, grouping senses of different words into the same explicit and comprehensive semantic class. Also the work presented by Mihalcea, Csomai, and Ciaramita (2007) makes use of three different sets of semantic classes (WordNet classes and two Named Entity annotated corpora) to train sequential classifiers. The classifiers are trained using basic features, collocations and semantic features, and they reach a performance around 60% and the 14th position in the SemEval-2007 all–words task.

The semantic classes of WordNet (also called SuperSenses) have been widely used in different works. For instance, Paaß and Reichartz (2009a) apply Conditional Random Fields to model the sequential context of words and their relation to SuperSenses. They also extend the model to include the potential SuperSenses of each word into the training data. An F1 score of 82.8% is reported (both nouns and verbs) when only potential labels are used (no training data at all) which is just 1% worse than when using the training data with right labels. Although interesting, they only evaluate the system applying a 5-fold cross validation on SemCor.

3. Semantic Classes and Levels of Abstraction

The meanings represented by WordNet have been only used for WSD at a very fine-grained sense level or at a very coarse-grained semantic class level (also called SuperSenses). We suspect that an appropriate level of abstraction could be found on between both levels. In this section we propose a

simple method to automatically derive semantic classes at intermediate levels of abstraction covering all nominal and verbal WordNet meanings. First, we introduce WordNet, the semantic resource and sense repository used by most WSD systems. Also note that all semantic classes used in our work are also linked to WordNet.

WordNet (Fellbaum, 1998) is an online lexical database of English which contains concepts represented by synsets, which are sets of synonyms of content words (nouns, verbs, adjectives and adverbs). One synset groups together several senses of different words which are synonyms.

In WordNet different types of lexical and semantic relations interlink different synsets, creating in this way a very large structured lexical and semantic network. The most important relation encoded in WordNet is the subclass relation (for nouns is called hyponymy relation and for verbs troponymy relation). Table 2 shows some basic figures of different WordNet versions including the total number of words, polysemous words, synsets, senses (all the possible senses for all the words) and average polysemy.

Version	Words	Polysemous	Synsets	Senses	Avg. Polysemy
WN 1.6	121,962	23,255	99,642	173,941	2.91
WN 1.7	144,684	24,735	109,377	192,460	2.93
WN 1.7.1	146,350	25,944	111,223	195,817	2.86
WN 2.0	152,059	26,275	115,424	203,145	2.94
WN 2.1	155,327	27,006	117,597	207,016	2.89
WN 3.0	155,287	26,896	120,982	206,941	2.89

Table 2: Statistics of WordNet versions.

3.1 SuperSenses

SuperSenses is the name of the WordNet Lexicographer Files within the framework of WSD[3]. More in detail, WordNet synsets are organized into forty five SuperSenses, based on syntactic categories (nouns, verbs, adjectives and adverbs) and logical groupings such as PERSON, PHENOMENON, FEELING, LOCATION, etc. There are 26 basic categories for nouns, 15 for verbs, 3 for adjectives and 1 for adverbs. In some cases, different senses of a word can be grouped at a high level under the same SuperSense, reducing the polysemy of the word. This is often the case of very similar senses of a word. Having just a few classes for adjectives and adverbs, SuperSense taggers have been usually developed only for nouns and verbs. (Tsvetkov, Schneider, Hovy, Bhatia, Faruqui, & Dyer, 2014) presents a very interesting study on tagging adjectives with SuperSenses acquired from GermaNet (Hamp, Feldweg, et al., 1997).

3.2 WordNet Domains

WordNet Domains[4] (WND) (Magnini & Cavaglià, 2000; L. Bentivogli & Pianta, 2004) is a hierarchy of 165 domains which have been used to label semi-automatically all WordNet synsets. This set of labels is organized into a taxonomy following the Dewey Decimal Classification System[5].

3. More information of these SuperSenses can be found at `http://wordnet.princeton.edu/wordnet/man/lexnames.5WN.html`.
4. `http://wndomains.itc.it`
5. `http://www.oclc.org/dewey`

For building WND, many labels were assigned to high levels of the WordNet hierarchy and were automatically inherited across the hypernym and troponym hierarchy. Thus, the semi-automatic method[6] used to develop this resource was not free of errors and inconsistencies (Castillo, Real, & Rigau, 2004; González, Rigau, & Castillo, 2012).

Information brought by domain labels is complementary to what is already in WordNet. WND present some characteristics that can be interesting for WSD. First of all, a domain label may contain senses from different WordNet sub–hierarchies (derived from different SuperSenses). For instance, the domain RELIGION contains senses such as *priest*, deriving from NOUN.PERSON and *church*, deriving from NOUN.ARTIFACT. Second, a domain label may also include synsets of different syntactic categories. For instance, the domain RELIGION also contains the verb *pray* or the adjective *holy*.

Furthermore, a single WND label can subsume different senses of the same word, reducing in this way its polysemy. For instance, the first and third senses of *church* in WordNet 1.6 have the domain label RELIGION.

3.3 SUMO Concepts

SUMO[7] (Niles & Pease, 2001) was created as part of the IEEE Standard Upper Ontology Working Group. Their goal was to develop a standard upper ontology to promote data interoperability, information search and retrieval, automated inference, and natural language processing. SUMO consists of a set of concepts, relations, and axioms that formalize an upper ontology. For these experiments, we used the complete WordNet 1.6 mapping with 1,019 SUMO labels (Niles & Pease, 2003). In this case, the three noun senses of *church* in WordNet 1.6 are classified as RELIGIOUSORGANIZATION, BUILDING and RELIGIOUSCEREMONY according to the SUMO ontology.

3.4 Example of Semantic Classes

As an example, table 3 presents the three senses and glosses of the word *church* in WordNet 1.6.

Sense	***WordNet 1.6***	
	word senses	**gloss**
1	church^1_n Christian_church^1_n Christianity^2_n	a group of Christians; any group professing Christian doctrine or belief: *church is a biblical term for assembly*
2	church^2_n church_building^1_n	for public (especially Christian) worship: *the church was empty*
3	church_service^1_n church^3_n	a service conducted in a church: *don't be late for church*

Table 3: Glosses and examples for the senses of *church*$_n$

In Table 4 we show the classes assigned to each sense according to the semantic resources introduced previously. For instance, considering WordNet Domains, it can be observed that the senses number 1 (group of Christians) and 3 (service conducted in a church) belong to the same domain

6. It was based on several cycles of manual checking over automatically labeled data.
7. http://www.ontologyportal.org

RELIGION. On the contrary, SuperSenses and SUMO represent the three senses of *church* using different semantic classes. Also note that the resulting assignment of semantic classes identifies each word sense individually.

	Semantic Class		
Sense	**SuperSense**	**WND**	**SUMO**
1	NOUN.GROUP	RELIGION	RELIGIOUSORGANIZATION
2	NOUN.ARTIFACT	BUILDINGS	BUILDING
3	NOUN.ACT	RELIGION	RELIGIOUSCEREMONY

Table 4: Semantic Classes for the noun $church_n$

3.5 Levels of Abstraction

Basic Level Concepts (Rosch, 1977) (hereinafter BLC) are the result of a compromise between two conflicting principles of characterization (general vs. specific):

- Represent as many concepts as possible
- Represent as many features as possible

As a result of this conflicting characterization, BLC typically should occur in the middle levels of the semantic hierarchies.

The notion of **Base Concepts** (hereinafter BC) was introduced in EuroWordNet (Vossen, 1998). BC are supposed to be the most important concepts in several language specific wordnets. This importance can be measured in terms of two main criteria:

- A high position in the semantic hierarchy
- Having many relations to other concepts

In EuroWordNet a set of 1,024 concepts were selected and called Common Base Concepts. Common BC are concepts that act as BC in at least two languages. Only local wordnets for English, Dutch and Spanish were used to select this set of Common BC. In later initiatives, similar sets have been derived to harmonize the construction of multilingual wordnets.

Considering both definitions, in the next section we present a method to automatically generate different sets of Basic Level Concepts from WordNet at different levels of abstraction.

4. Automatic Selection of Basic Level Concepts

Several approaches have been developed trying to alleviate the fine granularity problem of WordNet senses by obtaining word sense groupings (Hearst & Schütze, 1993; Peters et al., 1998; Mihalcea & Moldovan, 2001; Agirre & de Lacalle, 2003; Navigli, 2006; Snow et al., 2007; Bhagwani, Satapathy, & Karnick, 2013). In most cases the approach consists on grouping different senses of the same word, resulting in a decrease of the polysemy. Obviously, when the polysemy is reduced the WSD task as a classification problem becomes easier, and a system using these coarse senses obtain better results than other systems using word senses. Other works have used predefined sets of semantic classes, mainly SuperSenses (Segond et al., 1997; Ciaramita & Johnson, 2003; Curran,

2005; Villarejo et al., 2005; Ciaramita & Altun, 2006; Picca, Gliozzo, & Ciaramita, 2008; Paaß & Reichartz, 2009b; Tsvetkov et al., 2014).

In this section, we describe a simple method to automatically create different sets of Basic Level Concepts from WordNet. The method exploits the nominal and verbal structure of WordNet. The basic idea is that synsets in WordNet having a high number of relations are important, and they could be candidates to be a BLC. To capture the relevance of a synset in WordNet we have considered two options:

1. *All*: the total number of relations encoded in WordNet for the synset

2. *Hypo*: the total number of the hyponymy relations of the synset

Our method follows a bottom–up approach exploiting the hypernymy chains of WordNet. For each synset, the process starts visiting the synsets in the hyperonymy chain and selecting (and stopping the walk for this synset) as its BLC the ancestor having the first local maximum considering the total number of relations (either *All* or *Hypo*)[8]. For synsets having more than one hyperonym, the method chooses the one with the higher number of relations to continue the process. This process ends with a preliminary set of candidate synsets selected as potential BLC.

Additionally, each synset selected as a potential BLC candidate must subsume (or represent) at least a certain number of descendant synsets. Thus, the minimum number of synsets a BLC must subsume is another parameter of the algorithm, and it is represented by the symbol λ. Candidate BLCs that do not reach this threshold are discarded, and their subsumed synsets are reassigned to other BLC candidate appearing in higher levels of abstraction.

Algorithm 1 presents the pseudo–code of the algorithm. The parameters of the algorithm are: a WordNet resource, the type of relations considered (*All* or *Hypo*), and the minimum number of concepts that must be subsumed by each BLC (λ). The algorithm has two phases. The first one selects the candidate BLC, following the bottom–up approach. The second phase discards the candidate BLC that do not satisfy the λ threshold.

Figure 1 shows a schema to illustrate the selection process. Each node represents a synset, and the edges represent the hyperonymy relations (for instance, A is the hyperonym of D, and D is the hyperonym of F). The number under each synset indicates its number of hyponymy relations.

The schema illustrates the selection process of BLC candidates for synset J using criterion *Hypo*. The process starts checking the hyperonym of J, which is F. F has two hyperonyms, B and D. The next synset visited in the hyperonymy chain of J is D since it has a higher number of hyponymy relations (three). Again the algorithm compares the number of relations of the hyperonym synset (D with three relations), with those from the previous synset (F with two). As the number is increasing the process continues. Now, the next node to visit is A. As the number of relations of A is two and the number for D is three, the process stops and the synset selected as BLC candidate for J is D.

Table 5 shows a real example of the selection process for the noun *church* in WordNet 1.6. The hyperonym chain and the number of relations encoded in WordNet (*All* criterion) are shown for each synset. The local maximum in the chain is marked in bold.

8. The algorithm starts by checking the first hyperonym of the synset, not the synset itself.

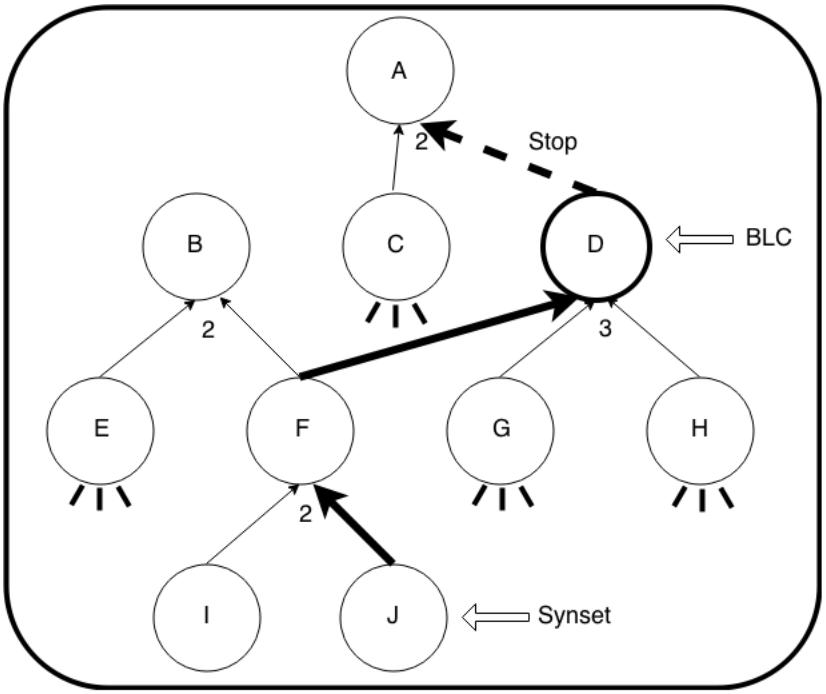

Figure 1: Example of BLC selection

#rel.	**synset**
18	group_1,grouping_1
19	social_group_1
37	*organisation_2,organization_1*
10	establishment_2,institution_1
12	faith_3,religion_2
5	Christianity_2,**church_1**,Christian_church_1

#rel.	**synset**
14	entity_1,something_1
29	object_1,physical_object_1
39	artifact_1,artefact_1
63	construction_3,structure_1
79	*building_1,edifice_1*
11	place_of_worship_1, ...
19	**church_2**,church_building_1

#rel.	**synset**
20	act_2,human_action_1,human_activity_1
69	*activity_1*
5	ceremony_3
11	religious_ceremony_1,religious_ritual_1
7	service_3,religious_service_1,divine_service_1
1	**church_3**,church_service_1

Table 5: BLC selection for the noun *church* in WordNet 1.6

Algorithm 1 BLC Extraction

Require: WordNet (WN) , typeOfRelation (T), threshold (λ)

$BlcCandidates = \emptyset$
for $each$ $\{synset\ S \in WN\}$ **do**
$cur := S$
{Obtaining the hypernym chains for the current synset *cur*}
$H := Hypernyms(WN, cur)$
$new := SynsetWithMoreRelations(WN, H, T)$
{Iterating while the number of relations is increased}
while $NumOfRels(WN, T, cur) < NumOfRels(WN, T, new)$ **do**
$cur := new$
$H := Hypernyms(WN, cur)$
$new := SynsetWithMoreRelations(WN, H, T)$
end while{Store *cur* as a candidate BLC}
$BlcCandidates := BlcCandidates \cup \{cur\}$
end for
{Filtering out the BLC candidates}
$BlcFinal = \emptyset$
for $each$ $\{blc \in BlcCandidates\}$ **do**
if $\lambda < NumberOfDescendants(WN, blc)$ **then**
$BlcFinal := BlcFinal \cup \{blc\}$
end if
end for
return $BlcFinal$

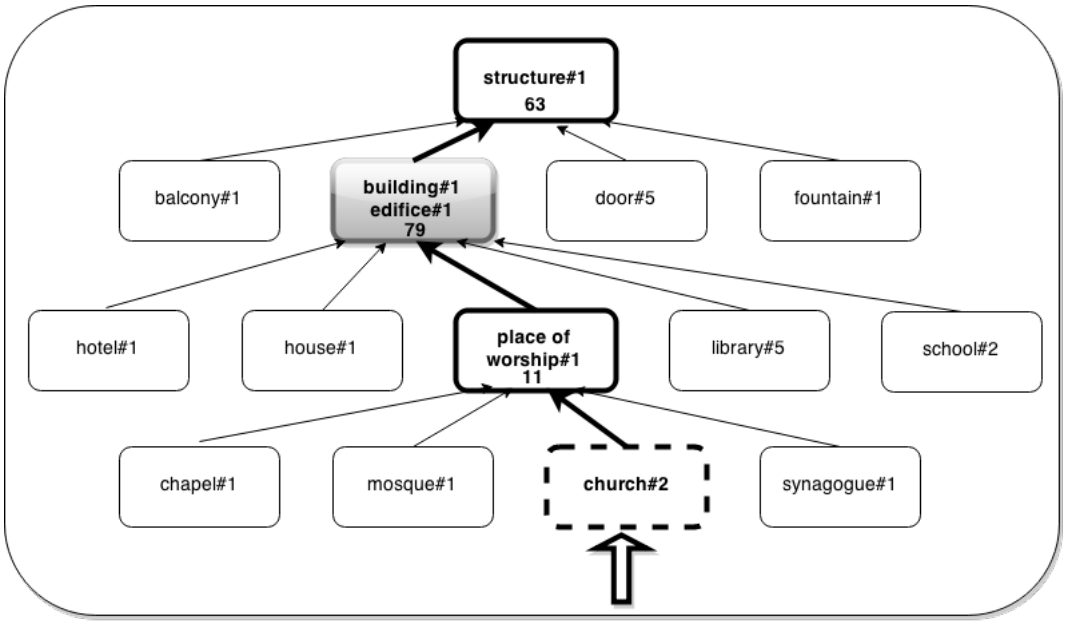

Figure 2: Example of BLC selection for the sense 2 of church

In figure 2 we can see a diagram showing a partial view of the selection process of a candidate BLC for the sense number 2 of the noun church. The synset dotted is the synset that is being processed (church^2_n). The synsets in bold are those that are visited by the algorithm, and the one in gray (building^1_n) is the one selected as BLC for church^2_n. The process stops checking the synset for structure^1_n as the number of relations is 63, which is lower than the number of relations of the previous synset (79 relations for edifice^1_n).

Obviously, combining different values for the λ threshold (for example 0, 10, 20 or 50) and the criterion considered by the algorithm (*All* or *Hypo*), the process ends up in different sets of BLC extracted automatically from any WordNet version.

Furthermore, instead of the number of relations we can consider the frequency of the synsets in a corpus as a measure of its importance. Synset frequency can be calculated as the sum of the frequencies of the word senses contained in the synset, which can be obtained from SemCor (Miller, Leacock, Tengi, & Bunker, 1993), or WordNet.

To sum up, the algorithm has two main parameters, the λ parameter, representing the minimum number of synsets that each BLC must represent, and the criterion used for characterizing the relevance of the synsets. The values for both parameters can be:

- λ parameter: any integer value greater or equal to 0
- Synset relevance parameter: the value considered to measure the importance of the synset. Four possibilities:
  - Number of relations of the synset
    - *All*: all relations encoded for the synset
    - *Hypo*: only hyponymy relations
  - Frequency of the synset
    - *FreqWN*: frequency obtained using WordNet
    - *FreqSC*: frequency obtained using SemCor

An implementation of this algorithm and the different sets of BLC used in this paper for several WordNet versions are freely available[9].

4.1 Analysis of Basic Level Concepts

We have selected WordNet 1.6 to generate several sets of BLC, combining the four types of synset *relevance criteria* and values 0, 10, 20 and 50 for λ. These values have been selected since they represent different levels of abstraction, ranging from $\lambda = 0$ (no filtering) to $\lambda = 50$ (each BLC must subsume at least 50 synsets). Table 6 shows, for combinations of λ and synset relevance parameters, the number of concepts that each set of BLC contains, and the average depth on the WordNet hierarchy of each group. In gray we highlight the two sets of BLC (BLC-20 and BLC-50 with "all_relations" parameter) that we use through all the experiments described in this paper.

As expected, increasing the λ threshold has a direct effect on the number of BLC and on its average depth in the WordNet hierarchy. In particular, both values are decreased, indicating that when the λ threshold is increased, the concepts selected are more abstract and general. For instance,

9. http://adimen.si.ehu.es/web/BLC

λ Threshold	Synset Relevance	# BLC		Depth	
		Nouns	Verbs	Nouns	Verbs
0	*All*	3,094	1,256	7.09	3.32
	Hypo	2,490	1,041	7.09	3.31
	FreqSC	34,865	3,070	7.44	3.41
	FreqWN	34,183	2,615	7.44	3.30
10	*All*	971	719	6.20	1.39
	Hypo	993	718	6.23	1.36
	FreqSC	690	731	5.74	1.38
	FreqWN	691	738	5.77	1.40
20	*All*	558	673	5.81	1.25
	Hypo	558	672	5.80	1.21
	FreqSC	339	659	5.43	1.22
	FreqWN	340	667	5.47	1.23
50	*All*	253	633	5.21	1.13
	Hypo	248	633	5.21	1.10
	FreqSC	94	630	4.35	1.12
	FreqWN	99	631	4.41	1.12

Table 6: Automatic Base Level Concepts for WN1.6

using (*All*) in the nominal part of WordNet, the number of concepts selected range from 3,094 with no filtering ($\lambda = 0$) to 253 ($\lambda = 50$). However, on average, its depth reduction is not so acute since it varies from 7.09 to 5.21. This fact shows the robustness of our method for selecting synsets from an intermediate level of abstraction.

Also as expected, the verbal part of WordNet behave differently. In this case, since the verbal hierarchies are less deep, the average depth of the synsets selected ranges from 3.32 to only 1.13 using *All* relations, and from 3.31 to 1.10 using *Hypo* relations.

In general, when using the frequency criteria, we can observe a similar behavior than when using the relation criteria. However, now the effect of the threshold is more dramatic, specially for nouns. Again, as expected, verbs behave differently than nouns. The number of BLC (for both SemCor and WordNet frequencies) reaches a plateau of around 600. In fact, this number is very close to the verbal top beginners of WordNet.

Summing up, we have devised a simple automatic procedure for deriving different sets of BLC representing at a different level of abstraction the whole set of nominal and verbal synsets of WordNet. In the following section we show and explain the supervised framework developed for WSD in order to exploit the semantic classes described in this section and the previous one.

5. Supervised Class-Based WSD

We follow a supervised machine learning approach to develop a set of semantic class based WSD classifiers. Our systems use an implementation of a Support Vector Machine algorithm to train the classifiers, one per semantic class, on semantic annotated corpora for acquiring both positive and negative examples of each class. These classifiers are built on the basis of a set of features defined for representing these examples. Being class-based, the training data must be collected and treated in a pretty different way than in the usual word-based approach.

First, word-based and class-based approaches selects the training examples very differently. In the word-based approach, only instances of the same word can be used as training examples. Figure 3 shows the distribution of training examples used to generate a word sense classifier for the noun *house*. Following the binary definition of SVM, one classifier is generated for each word sense. For each of these classifiers, only occurrences of the word sense associated with the classifier can be used as positive examples, while the rest of word sense occurrences are used as negative examples.

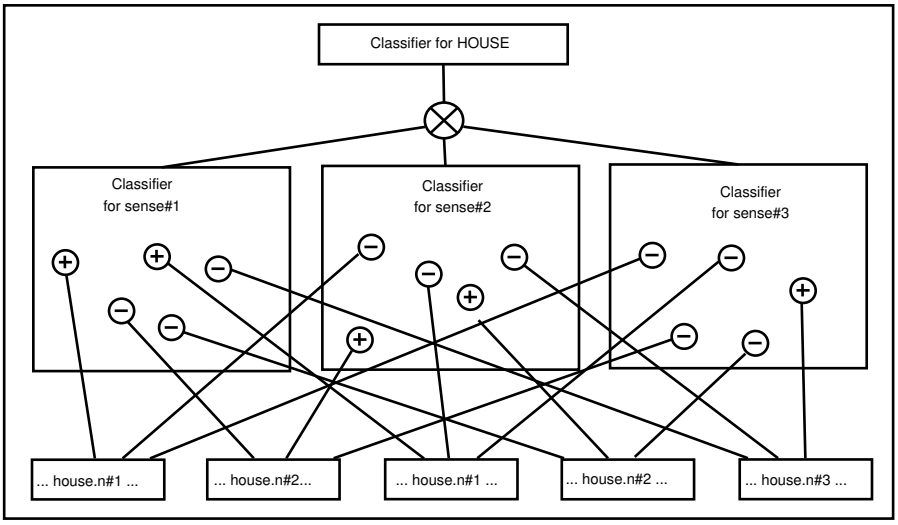

Figure 3: Distribution of examples using a word-based approach

In a class-based approach, we can use all the examples from all the words that belong to a particular semantic class. Figure 4 shows the distribution of examples in the class-based approach. In this case, one classifier is created for each semantic class. All occurrences of words belonging to the semantic class associated with the classifier can be used as positive examples, while the rest of occurrences of word senses associated with a different semantic class are selected as negative examples.

Obviously, in the class-based approach the number of examples for training is increased. Table 7 shows an example for sense $church_n^2$. Following a word-based approach only 58 examples can be found in Semcor for $church_n^2$. Conversely, 371 positive training examples can be used when building a classifier for the semantic class "building, edifice".

We think that this approach has several advantages. First, semantic classes reduce the average polysemy degree of words (some word senses might be grouped together within the same semantic class). Moreover, the acquisition bottleneck problem in supervised machine learning algorithms is attenuated because of the increase in the number of training examples. However, we are mixing in one classifier examples from very different words. For instance, for the *building* class we are grouping together examples from *hotel*, *hospital* or *church*, which could introduce noise in the learning process when grouping unrelated word senses.

5.1 The Learning Algorithm: SVM

Support Vector Machines (SVM) have been proven to be robust and very competitive in many NLP tasks, and in WSD in particular (Màrquez et al., 2006). In our experiments, we used SVM-Light

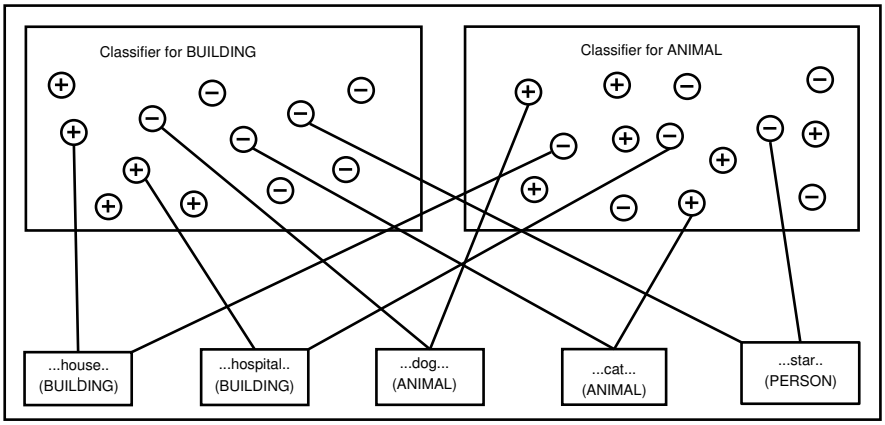

Figure 4: Distribution of examples using a class-based approach

Classifier	**Examples**	**# of positive examples**
church^2_n *(word-based approach)*	church^2_n	58
building, edifice *(class approach)*	church^2_n	58
	building^1_n	48
	hotel^1_n	39
	hospital^1_n	20
	barn^1_n	17
	.....	.....
		371 examples

Table 7: Number of examples in Semcor: word vs. class-based approaches

implementation (Joachims, 1998). SVM are used to induce a hyperplane that separates the positive from the negative examples with a maximum margin. It means that the hyperplane is located in an intermediate position between positive and negative examples, trying to keep the maximum distance to the closest positive example, and to the closest negative example. In some cases, it is not possible to get a hyperplane that divides the space linearly, or it is better to allow some errors to obtain a more efficient hyperplane. This is known as *soft-margin SVM*, and requires the estimation of a parameter (C), that represents the trade-off allowed between training errors and the margin. We have set this value to 0.01, which has been demonstrated as a good value for SVM in WSD tasks.

When classifying an example, we obtain the value of the output function for each SVM classifier corresponding to each semantic class for the word example, and our system simply selects the class having the greatest value.

5.2 Corpora

Three semantic annotated corpora have been used for training and testing. Semcor for training, and SensEval-2 and SensEval-3 English all-words tasks, for testing.

SemCor (Miller et al., 1993) is a subset of the Brown Corpus plus the novel *The Red Badge of Courage*, and it has been developed by the same group that created WordNet. It contains 253 texts and around 700,000 running words, and more than 200,000 are also lemmatized and sense-tagged according to Princeton WordNet 1.6. The sense annotations from SemCor have been also automatically ported to other WordNet versions[10].

SensEval-2[11] English all-words corpus (hereinafter SE2) (Palmer, Fellbaum, Cotton, Delfs, & Dang, 2001) consists of 5,000 words of text from three Wall Street Journal (WSJ) articles representing different domains from the Penn TreeBank II. The sense inventory used for tagging was WordNet 1.7.

SensEval-3[12] English all-words corpus (hereinafter SE3) (Snyder & Palmer, 2004), is made up of 5,000 words, extracted from two WSJ articles and one excerpt from the Brown Corpus. Sense repository of WordNet 1.7.1 was used to tag 2,041 words with their proper senses.

We also considered alternative evaluation datasets. For instance, the SemEval-2007 coarse–grained task corpus[13]. However, this dataset has been discarded because this corpus is annotated with a particular set of word sense clusters. Additionally, it does not provide a clear and simple way to compare orthogonal sets of clusterings. Although there have been more recent SensEval/SemEval tasks about WSD, we think that for the purpose of this evaluation (different level of abstraction in WSD), SensEval-2 and SensEval-3 are still the datasets that best fit to our purposes. More recent SemEval competitions have been designed to address specific topics, such as multilinguality or joint WSD and Named Entity Recognition. However, we have also make some additional experiments on domain adaptation with the dataset provided by SemEval-10 task 17 "All-words Word Sense Disambiguation on a Specific Domain (WSD-domain)"[14] (Agirre, López de Lacalle, Fellbaum, Hsieh, Tesconi, Monachini, Vossen, & Segers, 2010).

10. http://web.eecs.umich.edu/~mihalcea/downloads.html#semcor
11. http://www.sle.sharp.co.uk/senseval2
12. http://www.senseval.org/senseval3
13. Indeed we participated in this task with a preliminary version of our system
14. http://semeval2.fbk.eu/semeval2.php?location=tasks#T25

5.3 Feature Types

Following previous contributions in supervised WSD, we have selected a set of basic features to represent the training and testing examples. We also include additional features based on semantic classes.

- **Basic features**
  - **Word-forms and lemmas** in a window of 10 words around the target word.
  - **PoS**, the concatenation of the preceding/following three and five PoS tags.
  - **Bigrams and trigrams** formed by lemmas and word-forms in a window of 5 words around the target word; we use all tokens regardless their PoS to build bi/trigrams. We replace the target word by a character 'X' in these features to increase its generalization.
- **Semantic features**
  - **Most frequent semantic class** for the target word, calculated over SemCor.
  - **Monosemous semantic class** of monosemous words in a window of size five words around the target word.

Basic features are those widely used in the literature, as the work presented by Yarowsky (1994). These features are pieces of information that occur in the context of the target word: local features including bigrams and trigrams (including the target word) of lemmas, word-forms or part–of–speech labels (PoS). In addition, word–forms or lemmas in some larger window around the target word are considered as features representing the topic of the discourse.

The set of features is extended with semantic information. Several types of semantic classes have been considered to create these features. In particular, two different sets of BLC (BLC20 and BLC50[15]), SuperSenses, WordNet Domains (WND) and SUMO.

In order to increase the generalization capabilities of the class-based classifiers we filter out irrelevant features. We measure the relevance of a feature[16] f for a class c in terms of the frequency of f. For each class c, and for each feature f of that class, we calculate the frequency of the feature within the class (the number of times that it occurs in examples of the class), and we also obtain the total frequency of the feature for all the classes. We get the relative frequency by dividing both values (classFreq / totalFreq) and if the result is lower than a certain threshold t, the feature is removed from the feature list of the class c[17]. In this way, we make sure that the features selected for a class are more frequently related with that class than with others. We set this threshold t to 0.25, obtained empirically with very preliminary versions of the classifiers when applying a cross-validation setting on SemCor.

15. We have selected these set since they represent different levels of abstraction. As said in section 4, 20 and 50 refer to the threshold of minimum number of synsets that a possible BLC must subsume to be considered as a proper BLC. These sets of BLC were built using *all* criterion.
16. That is, the value of the feature, for example a *feature type* can be **word-form**, and a *feature* of that type can be *houses*.
17. Depending on the experiment, around 30% of the original features are removed by this filter.

6. Semantic Class–Based WSD Experiments

In this section we present the performance of our semantic class-based WSD system in the all–words WSD SensEval-2 (SE2) and SensEval–3 (SE3) datasets. We want to analyze the behavior of our class-based WSD system when working at different levels of abstraction. As we have said before, the level of abstraction is defined by the semantic class used to build the classifiers.

An experiment is defined by two different parameters each one involving a particular set of semantic classes.

1. **Target class:** The semantic classes used to train the classifiers (determining the abstraction level of the system). In this case, we tested: word-sense[18], BLC20, BLC50, WordNet Domains (WND), SUMO and SuperSenses (SS).

2. **Semantic features class:** The semantic classes used for building the semantic features. In this case, we tested: BLC20, BLC50, WND, SUMO and SuperSenses (SS).

The target class is the type of classes that the classifier assigns to a given ambiguous word. For instance, the target class for the traditional word expert classifiers are word senses. The Semantic feature class is the one used for building the semantic features, which is independent of the target class. For instance, we can use WordNet Domains to extract monosemous words from the context of the target word and use the WND labels of these words as semantic features for building the classifier.

Combining different semantic classes for target and features, we generated the set of experiments described in the next sections. In that way, we can evaluate independently the impact of selecting one semantic class or another as target class or as semantic feature class.

Test	PoS	Sense	BLC20	BLC50	SUMO	SS	WND
SE2	N	4.02	3.45	3.34	3.33	2.73	2.66
	V	9.82	7.11	6.94	5.94	4.06	2.69
SE3	N	4.93	4.08	3.92	3.94	3.06	3.05
	V	10.95	8.64	8.46	7.60	4.08	2.49

Table 8: Average polysemy on SE2 and SE3

Table 8 shows the average polysemy (AP) measured on SE2 and SE3 with respect to the different semantic classes used in our evaluation as target classes. As expected, every corpus behaves differently and the average polysemy for verbs is higher than for nouns. Also as we could assume in advance, relevant reductions on the polysemy degree are obtained when increasing the level of abstraction. This fact is more acute also for verbs. Note the large reduction of polysemy for verbs when using SuperSenses and also WND. Also note that *a priori* SE3 seems to be more difficult to disambiguate than SE2, independently of its abstraction level.

6.1 Baselines

As baselines of these evaluations we define the most frequent classes (MFC) of each word calculated over SemCor. Ties between classes on a specific word are solved obtaining the global frequency in

18. We included a word-based evaluation for comparison purposes only since the current system have been designed for class-based evaluation.

SemCor of each of these tied classes, and selecting the most frequent class over the whole training corpus. When Semcor has no occurrences of a particular word (that is, we are not able to calculate the most frequent class of the word), we compute the global frequency for each of its possible semantic classes (obtained from WordNet) over SemCor, and we select the most frequent one. Table 9 shows the baseline for each semantic class over both testing corpora.

Class	Pos	SE2		SE3	
		MFC	AP	MFC	AP
Sense	N	70.02	4.02	72.30	4.93
	V	44.75	9.82	52.88	10.95
BLC20	N	75.71	3.45	76.29	4.08
	V	55.13	7.11	58.82	8.64
BLC50	N	76.65	3.34	76.64	3.92
	V	54.93	6.94	60.05	8.46
SUMO	N	76.09	3.33	79.55	3.94
	V	60.35	5.94	64.71	7.60
SuperSense	N	80.41	2.73	81.50	3.06
	V	68.47	4.06	79.07	4.08
WND	N	86.11	2.66	83.82	3.05
	V	90.33	2.69	92.20	2.49

Table 9: Most Frequent Class baselines and average polysemy (AP) on SE2 and SE3

As expected, the performances of the MFC baselines are very high. In particular, those corresponding to nouns (ranging from 70% to 80%). While nominal baselines seem to perform similarly in both SE2 and SE3, verbal baselines appear to be consistently much lower for SE2 than for SE3. In SE2, verbal baselines range from 44% to 68% while in SE3 verbal baselines range from 52% to 79%. The results of WND are very high due to its low polysemy degree for both nouns and verbs. Obviously, when increasing the level of abstraction (from senses to WND) the results also increase.

6.2 Results of Our Basic System

In this section we present the performance of our supervised semantic class–based WSD system. Table 10 shows the results of the system when trained varying the target classes and using only the basic feature set. Their values correspond to the F1 measures (harmonic mean of recall and precision) when training our systems on SemCor and testing on SE2 and SE3 test sets. The results that improve the baselines are shown in italics. Additionally, those results showing a statistically significant positive difference when compared with its corresponding baseline using McNemar's test are marked in bold.

Interestingly, only the basic system at a word-sense level outperforms the baselines in SE2 and SE3 for both nouns and verbs. In addition, our systems obtain in some cases significantly better results for verbs. Also interesting is that on verbs at a word-sense level the baselines and results are very different, while at a class-level the differences on both datasets are much smaller.

As expected, the results of the systems increase when augmenting the level of abstraction (from senses to WND), and in most cases, the baseline results are reached or outperformed. This is even more relevant if we consider that the baseline results are already quite high. However, at a very high level of abstraction (SuperSenses or WND) our basic systems seem to be unable to outperform the baselines.

Class	Pos	SE2	SE3
Sense	N	*71.20*	*73.15*
	V	*45.53*	***57.02***
BLC20	N	75.52	73.82
	V	*57.06*	***61.10***
BLC50	N	74.57	*75.84*
	V	***58.03***	*61.97*
SUMO	N	*77.60*	76.74
	V	*62.09*	*66.21*
SuperSense	N	79.94	79.48
	V	***71.95***	78.39
WND	N	80.81	77.64
	V	90.14	88.92

Table 10: Results of the basic system trained on SemCor with a basic set of features and evaluated against SE2 and SE3

In general, the results obtained by BLC20 are not so different to those of BLC50. For instance, if we consider the number of classes within BLC20 (558 classes), BLC50 (253 classes) and SuperSense (24 classes), BLC classifiers obtain high performance rates while maintaining much higher expressive power than SuperSenses (they are able to classify among much larger number of classes). In fact, using SuperSenses (40 classes for nouns and verbs) we obtain a very accurate semantic tagger with a performance close to 80%. Even more interesting, we could use BLC20 for tagging nouns (558 semantic classes and F1 around 75%) and SuperSenses for verbs (14 semantic classes and F1 around 75%).

6.3 Results Exploiting the Semantic Features

One of our main goals is to prove that simple semantic features added to the training process are capable of producing significant improvements against the basic systems. The results of the experiments considering also the different types of semantic features are presented in Tables 11 and 12, respectively for nouns and verbs.

In both tables, the column labeled as *Class* refers to what we have called the target class, and the column labeled as *SF* indicates the type of semantic features included to represent the examples within our machine learning approach.

Again, the values in the tables correspond to the F1 measures (harmonic mean of recall and precision) when training our systems on SemCor and testing on SE2 and SE3 test sets. The results improving the baselines appear in italics. Additionally, those results showing a statistically significant positive difference when compared with its corresponding baseline using the McNemar's test are marked in bold.

Regarding nouns (see Table 11), a very different behavior is observed for SE2 and SE3. Adding semantic features mainly improves the results on SE2. While for SE3 none of the systems present a significant improvement over the baselines, for SE2 such improvement is obtained when using several types of semantic features (in particular, when using WND features on SE2). The use of semantic class-based features seems to improve the systems using as target classes intermediate levels of abstraction (specially BLC20 and BLC50). Interestingly, in SE3 only BLC20 and BLC50

Class	SF	SE2	SE3	Class	SF	SE2	SE3
Sense	baseline	70.02	72.30	SUMO	baseline	76.09	79.55
	basicFeat	*71.20*	*73.15*		basicFeat	*77.60*	76.74
	BLC20	*71.79*	*73.15*		BLC20	75.52	76.74
	BLC50	*71.69*	*73.04*		BLC50	75.52	77.19
	SUMO	*71.59*	*73.15*		SUMO	*77.88*	78.76
	SS	*71.10*	*72.70*		SS	*77.50*	76.97
	WND	*71.20*	*73.15*		WND	***77.88***	77.42
BLC20	baseline	75.75	76.29	SS	baseline	80.41	81.50
	basicFeat	75.52	73.82		basicFeat	79.94	79.48
	BLC20	***77.69***	*76.52*		BLC20	*81.07*	81.39
	BLC50	***77.79***	75.73		BLC50	80.22	*81.73*
	SUMO	*77.60*	73.71		SUMO	*80.51*	81.05
	SS	75.14	73.82		SS	80.32	76.46
	WND	***77.88***	74.24		WND	***82.47***	79.82
BLC50	baseline	76.65	76.74	WND	baseline	86.11	83.82
	basicFeat	74.57	75.84		basicFeat	80.81	77.64
	BLC20	***78.45***	*76.85*		BLC20	81.85	80.79
	BLC50	*76.65*	*76.74*		BLC50	82.33	80.11
	SUMO	***79.58***	75.51		SUMO	83.55	81.24
	SS	75.52	74.61		SS	83.08	78.31
	WND	***78.92***	74.83		WND	86.01	83.71

Table 11: Results for **nouns** using the extended system

seem to provide some improvements over the baselines in some of the target classes (for instance, BLC20, BLC50 and SS), although not significant.

Regarding verbs (see Table 12), also a very different behavior is observed for SE2 and SE3. In this case, we can observe almost the opposite effect than for nouns. On SE3 most of the semantic class features improve the results obtained by the baselines. While for SE2 only some of the systems present a significant improvement over the baselines, for SE3 such improvement is obtained when using several types of semantic features. However, in this case we also obtain significantly better results for several semantic features on SE2. The use of semantic class-based features seems to benefit lower levels of abstraction (specially word-sense, BLC20, BLC50 and also SUMO).

In general, the results show that using semantic features in addition to the basic features helps to reach a better performance for the class-based WSD systems. Additionally, it also seems that using these semantic features we are able to obtain very competitive classifiers at a sense level.

6.4 Learning Curves

We also investigate the behavior of the class-based WSD system with respect the number of training examples. Although the same experiments have been carried out for nouns and verbs, we only include the results for nouns since in both cases, the trend is very similar.

In these experiment, the Semcor files have been randomly selected and added to the training corpus in order to generate subsets of 5%, 10%, 15%, etc. of the training corpus[19]. Then, we train

19. Each portion contains also the same files than the previous portion. For example, all files in the 25% portion are also contained in the 30% portion.

Class	SF	SE2	SE3	Class	SF	SE2	SE3
Sense	baseline	44.75	52.88	SUMO	baseline	60.35	64.71
	basicFeat	*45.53*	***57.02***		basicFeat	*62.09*	*66.21*
	BLC20	*45.14*	***56.61***		BLC20	*61.12*	*66.07*
	BLC50	*45.53*	***56.47***		BLC50	*62.09*	*66.48*
	SUMO	*45.73*	***57.02***		SUMO	*60.74*	*64.98*
	SS	*45.34*	***56.75***		SS	59.96	*64.71*
	WND	*45.53*	***56.75***		WND	*61.51*	*66.35*
BLC20	baseline	55.13	58.82	SS	baseline	68.47	79.07
	basicFeat	*57.06*	***61.10***		basicFeat	***71.95***	78.39
	BLC20	*56.87*	*59.92*		BLC20	*69.25*	77.70
	BLC50	*55.90*	***60.60***		BLC50	*69.25*	77.70
	SUMO	*57.06*	***61.15***		SUMO	*70.21*	77.70
	SS	*56.29*	***61.29***		SS	*69.25*	77.84
	WND	***58.61***	*60.88*		WND	***71.76***	*79.75*
BLC50	baseline	54.93	60.05	WND	baseline	90.33	92.20
	basicFeat	***58.03***	*61.97*		basicFeat	90.14	88.92
	BLC20	***57.45***	*61.29*		BLC20	90.14	90.42
	BLC50	*56.67*	*61.01*		BLC50	90.14	90.15
	SUMO	*57.06*	*61.83*		SUMO	*90.52*	89.88
	SS	*57.45*	*61.83*		SS	89.75	88.78
	WND	***59.77***	***62.38***		WND	*90.52*	*92.20*

Table 12: Results for **verbs** using the extended system

the system on each of the training portions and we test the system on SE2 and SE3. Finally, we also compare the resulting system with the baseline computed over the same training portion.

Figures 5 and 6 present the learning curves over SE2 and SE3, respectively. In this case, we selected a BLC20 class-based WSD system using WordNet Domains as semantic features[20].

Surprisingly, in SE2 the system only improves the F1 measure around 2% while increasing the training corpus from 25% to 100% of SemCor. In SE3, the system again only improves the F1 measure around 3% while increasing the training corpus from 30% to 100% of SemCor. That is, most of the knowledge required for the class-based WSD system seems to be already present on a small part of SemCor.

Figures 7 and 8 present the learning curves over SE2 and SE3, respectively, of a class-based WSD system based on SuperSenses using as semantic features those built with WordNet Domains.

In SE2 the system just improves the F1 measure around 2% while increasing the training corpus from 25% to 100% of SemCor. In SE3, the system again only improves the F1 measure around 2% while increasing the training corpus from 30% to 100% of SemCor. That is, with only 25% of the whole corpus, the class-based WSD system reaches a F1 close to the performance using all corpus.

In SE2 ans SE3, when using BLC20 (Figures 5 and 6) or SuperSenses (Figures 7 and 8) as semantic classes for WSD, the behavior of the system is similar to the MFC baseline. This is very interesting since the MFC obtains very high results due to the way it is defined: the MFC over the total corpus is assigned if there are no occurrences of the word in the training corpus. Without this definition, there would be a large number of words in the test set with no occurrences when using

20. As shown in previous experiments, this combination obtains a very good performance.

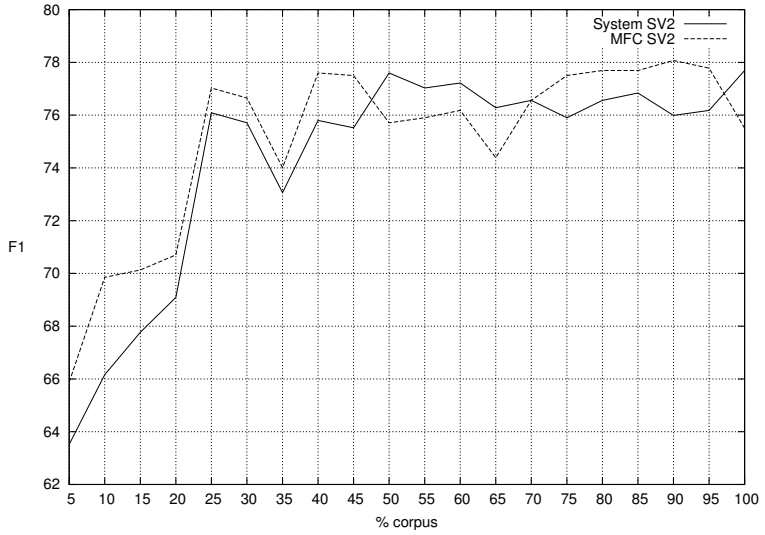

Figure 5: Learning curve of BLC20 classifier on SE2

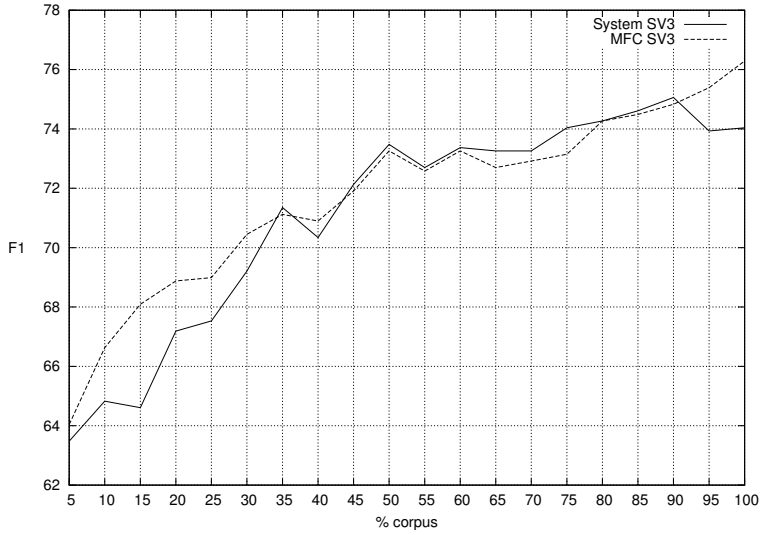

Figure 6: Learning curve of BLC20 classifier on SE3

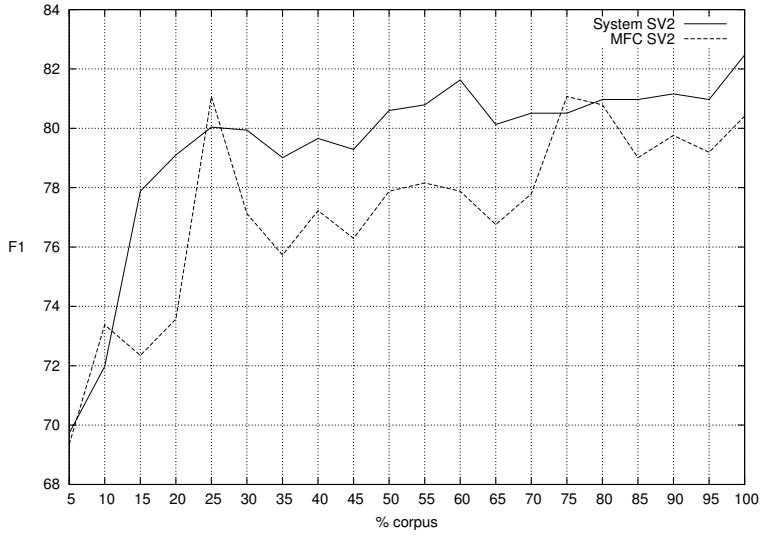

Figure 7: Learning curve of SuperSense classifier on SE2

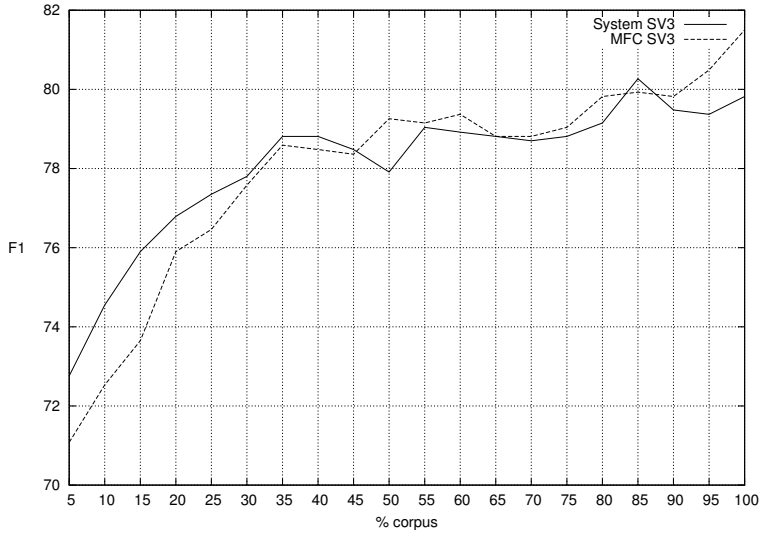

Figure 8: Learning curve of SuperSense classifier on SE3

small training portions. In these cases, the recall of the baselines (and in turn F1) would be much lower.

This evaluation seems to indicate that the class-based approach to WSD reduces considerably the required amount of training examples.

7. Comparison with SensEval Systems: Sense Level

The main goal of the experiments included in this section is to verify whether the abstraction level of our class-based systems maintains its discriminative power when evaluated at a sense level. Additionally, we compare our results against the results of the top participant systems in SE2 and SE3 which provided the best sense–level outputs. Thus, our class-based systems have been adapted following a simple protocol. The output based on semantic classes is converted to sense identifiers: instead of the semantic class produced by our systems for a particular instance, we select the first sense of the word according to the WordNet sense ranking belonging to the predicted semantic class. So, first we obtain the semantic class by means of our classifiers, then we obtain the restricted set of senses for the word that match the semantic class obtained, and then we choose the most frequent sense from that restricted subset.

The results of the first experiment on SE2 data are shown in Table 13. All our systems have the prefix "SVM-" while the suffix denotes the type of semantic class used to generate the classifier[21]. In all cases in these experiments, WND has been selected as target semantic class to generate the semantic features. Two baselines marked in Italics have been also included. The first sense in WordNet (*base-WordNet*) and the most frequent sense in SemCor (*base-SemCor*). In fact, the developers of WordNet ranked their word senses using SemCor and other sense-annotated corpora. Thus, the frequencies and ranks appearing in SemCor and in WordNet are similar, but not equal. We also include the results of our system when working at a word level (SVM-sense).

In both cases, for nouns and verbs, our systems outperform the most frequent baselines. The most frequent sense for a word, according to the WordNet sense ranking is very competitive in WSD tasks, and it is extremely hard to improve upon even slightly (McCarthy, Koeling, Weeds, & Carroll, 2004). As expected, the behavior of the different semantic features produces slightly different results. However, independently of the semantic features used, in SE2 at a sense level, the class-based systems rank at the third position.

Table 14 shows the same experiment but using SE3 dataset. In this case, our class-based systems clearly outperform the baselines, achieving the best results for nouns and the second place for verbs. Interestingly, for nouns, the best system at the SE3 did not achieve the SemCor baseline. Also recall that SE3 seems to be more difficult than SE2.

It is worth to mention that our class-based systems use the same features for both nouns and verbs. For instance, we do not take profit of complex feature sets encoding syntactic information that seems to be important for verbs.

These experiments show that class-based classifiers seem to be quite competitive when evaluated at a word sense level. They perform over the most frequent sense according to WordNet and SemCor, and achieve the higher position for nouns and the second for verbs in SE3, and the third position for nouns and verbs in SE2. Obviously, this indicates that class-based WSD maintains a very high discriminative power at a word sense level.

21. For instance, *SVM-BLC20* stands for the experiment that creates classifier considering BLC20 semantic classes.

Class → Sense on SE2			
Nouns		**Verbs**	
System	**F1**	**System**	**F1**
SMUam	73.80	SMUaw	52.70
AVe-Antwerp	74.40	AVe-antwerp	47.90
SVM-semBLC20	71.80	SVM-semSUMO	45.70
SVM-semBLC50	71.70	SVM-sense	45.53
SVM-semSUMO	71.60	SVM-semWND	45.50
SVM-semWND	71.20	SVM-semBLC50	45.50
SVM-sense	71.20	SVM-semSS	45.30
SVM-semSS	71.10	SVM-semBLC20	45.10
base-WordNet	*70.10*	LIA-Sinequa	44.80
base-SemCor	*70.00*	*base-SemCor*	*44.80*
LIA-Sinequa	70.00	*base-WordNet*	*43.80*

Table 13: Class to Sense results on SE2. Class to word sense transformation.

Class → Sense on SE3			
Nouns		**Verbs**	
System	**F1**	**System**	**F1**
SVM-semWND	73.20	GAMBL-AW	59.30
SVM-semBLC20	73.20	SVM-semSUMO	57.00
SVM-semSUMO	73.20	SVM-semWND	56.80
SVM.sense	73.15	SVM-semSS	56.80
SVM-semBLC50	73.00	SVM-sense	56.75
SVM-semSS	72.70	SVM-semBLC20	56.60
base-SemCor	*72.30*	SVM-semBLC50	56.50
GAMBL-AW	70.80	UNTaw	56.40
base-WordNet	*70.70*	Meaning-allwords	55.20
kuaw	70.60	kuaw	54.50
UNTaw	69.60	R2D2	54.40
Meaning-allwords	69.40	*base-SemCor*	*52.90*
LCCaw	69.30	*base-WordNet*	*52.80*

Table 14: Class to Sense results on SE3. Class to word sense transformation.

8. Comparison with SensEval Systems: Class Level

The experiments presented in this section explore the performance of the word-based classifiers participating at SE2 and SE3 when are evaluated at a class level. To perform this kind of evaluation, the word sense output of the participant systems have been mapped to its corresponding semantic classes. Our class-based systems are not modified. Obviously, we expect different performances of the systems depending on the semantic class level. Considering the results presented in tables 11 and 12, in order to perform the comparison, we have selected the experiments that use WND to build the semantic features[22]. Thus, our system results using the different target semantic classes are all represented by *SVM-semWND*.

Table 15 presents ordered by F1-measure the results of the best performing systems on SE2 data when evaluated at a different levels of abstraction. As previously, in italics we include the most frequent senses according to WordNet *base-WordNet* and SemCor *base-SemCor*.

On SE2, independently of the abstraction level and PoS, our system (SVM-semWND) scores at the first positions of the ranking. In one case our system reaches the best position, and twice the second one. The baselines are outperformed in all experiments, except for nouns using WND, where *base–SemCor* is very high.

Table 16 presents ordered by F1-measure the results of the best performing systems on SE3 data when evaluated at a different levels of abstraction. In italics we include the most frequent senses according to WordNet *base-WordNet* and SemCor *base-SemCor*. Our systems are again represented by *SVM-semWND*.

On SE3, we can see that our system performs better than the baselines in most cases, except for the SemCor–based baseline on nouns, which obtains a very high result. In particular, our system obtains very good results on verbs, reaching the first or second best positions in all cases, and outperforming both baselines in all cases.

To sum up, our class–based approach outperforms most SensEval participants (both SE2 and SE3), at sense level and at semantic class level. This suggests that the good performance of the semantic classifiers are not only due to the polysemy reduction. Actually, it confirms that our class–based semantic classifiers are learning from the semantic class training examples at different abstraction levels.

9. Out of Domain Evaluation

In this section we describe our system at the SemEval-2 *"All–words Word Sense Disambiguation on a Specific Domain"* task (Izquierdo, Suárez, & Rigau, 2010). The aim of this evaluation is to show how robust our semantic class approach is when tested on a specific domain, different to the domain of the training material.

Traditionally, SensEval competitions have been focused on general domain texts. Thus, domain specific texts present fresh challenges for WSD. For example, specific domains reduce the possible meaning of a word in a given context. Moreover, the distribution of word senses on the data examples changes when compared to general domains. These problems affect both supervised and knowledge–based systems. In fact, supervised word-based WSD systems are very sensitive to the corpora used for training and testing the system (Escudero et al., 2000).

22. Remind that the semantic features are the most frequent class of the target word, and the semantic class of monosemous words in the context around target word.

Nouns		Verbs	
System	**F1**	**System**	**F1**
Sense → BLC20			
SMUaw	78.72	SMUaw	61.22
SVM-semWND	77.88	SVM-semWND	58.61
AVe-antwerp	76.71	LIA-Sinequa	57.42
base-SemCor	75.71	AVe-antwerp	57.28
base-WordNet	74.29	*base-SemCor*	*55.13*
LIA-Sinequa	73.39	*base-WordNet*	*54.16*
Sense → BLC50			
SMUaw	79.01	SMUaw	61.61
SVM-semWND	78.92	SVM-semWND	59.77
AVe-antwerp	77.57	LIA-Sinequa	57.81
base-SemCor	*76.65*	AVe-Antwerp	57.67
base-WordNet	*75.24*	*base-SemCor*	*54.93*
LIA-Sinequa	74.53	*base-WordNet*	*54.55*
Sense → SUMO			
SMUaw	79.30	SMUaw	68.22
SVM-semWND	77.88	LIA-Sinequa	64.79
base-SemCor	*76.09*	AVe-Antwerp	62.56
AVe-Antwerp	75.94	SVM-semWND	61.51
LIA-Sinequa	*74.92*	*base-SemCor*	*61.33*
base-WordNet	*71.74*	*base-WordNet*	*60.35*
Sense → SuperSense			
SVM-semWND	82.47	SMUaw	73.47
SMUaw	81.21	LIA-Sinequa	72.74
AVe-Antwerp	80.75	SVM-semWND	71.76
base-SemCor	*80.41*	AVe-Antwerp	69.31
LIA-Sinequa	79.58	*base-WordNet*	*69.05*
base-WordNet	*78.16*	*base-SemCor*	*68.47*
Sense → WND			
SMUaw	88.80	SMUaw	91.16
base-SemCor	*86.11*	SVM-semWND	90.52
SVM-semWND	86.01	*base-SemCor*	*90.33*
AVe-Antwerp	87.30	LIA-Sinequa	89.82
base-WordNet	*85.82*	*base-WordNet*	*89.75*
LIA-Sinequa	84.85	AVe-Antwerp	89.74

Table 15: Results for sense to BLC20, BLC50, SUMO, SuperSense and WND semantic classes on SE2

Nouns		Verbs	
System	**F1**	**System**	**F1**
Sense → BLC20			
base-SemCor	*76.29*	GAMBL-AW	63.56
GAMBL-AW	74.77	SVM-semWND	60.88
kuaw	74.69	kuaw	60.66
LCCaw	74.44	R2D2	59.79
UNTaw	74.40	UNTaw	59.73
SVM-semWND	74.24	Meaning-allwords	59.37
base-WordNet	*74.16*	*base-SemCor*	*58.82*
Meaning-allwords	73.11	*base-WordNet*	*58.28*
Sense → BLC50			
base-SemCor	*76.74*	GAMBL-AW	64.38
GAMBL-AW	75.56	SVM-semWND	62.38
kuaw	75.25	kuaw	61.22
SVM-semWND	74.83	R2D2	60.35
LCCaw	74.78	UNTaw	60.27
UNTaw	74.73	Meaning-allwords	60.19
base-WordNet	*74.49*	*base-SemCor*	*60.06*
R2D2	73.93	*base-WordNet*	*58.82*
Sense → SUMO			
base-SemCor	*79.55*	GAMBL-AW	68.77
kuaw	78.18	SVM-semWND	66.35
LCCaw	77.54	UNTaw	66.03
SVM-semWND	77.42	kuaw	65.93
UNTaw	77.32	Meaning-allwords	65.43
GAMBL-AW	77.14	upv-eaw2	64.92
base-WordNet	*76.97*	*base-SemCor*	*64.71*
Meaning-allwords	76.75	*base-WordNet*	*64.02*
Sense → SuperSense			
base-SemCor	*81.50*	SVM-semWND	79.75
kuaw	79.89	GAMBL-AW	79.40
SVM-semWND	79.82	*base-SemCor*	*79.07*
UNTaw	79.71	*base-WordNet*	*78.25*
GAMBL-AW	79.62	Meaning-allwords	78.14
upv-eaw2	79.27	Meaning-simple	77.72
upv-eaw	78.42	kuaw	77.53
base-WordNet	*78.25*	upv-eaw2	77.21
Sense → WND			
base-SemCor	*83.80*	SVM-semWND	92.20
SVM-semWND	83.71	*base-SemCor*	*92.20*
UNTaw	83.62	UNTaw	91.37
kuaw	81.78	GAMBL-AW	91.01
GAMBL-AW	81.53	*base-WordNet*	*90.83*
base-WordNet	*81.46*	R2D2	90.52
LCCaw	80.64	Meaning-simple	90.50
Meaning-allwords	80.50	kuaw	90.44

Table 16: Results for sense to BLC20, BLC50, SUMO, SuperSense and WND semantic classes on SE3

Therefore, the main challenge is how to develop specific domain WSD systems or how to adapt a general system to a particular domain. Following this research line, a task was proposed within the SemEval–2 competition: *"All–words Word Sense Disambiguation on a Specific Domain"* (Agirre et al., 2010). The restricted domain selected for this task was the environmental domain. The test corpora consist of three texts compiled by the European Center for Nature Conservation[23] (ECNC) and World Wildlife Forum[24] (WWF). The task was proposed in several languages: Chinese, Dutch, English an Italian, although our participation was limited to English. More in detail, there were a total of 1,032 noun tokens and 366 verb tokens to be tagged. Moreover, a set of background documents related with the environmental domain were provided. These texts were no sense tagged, they were just plain text, and they were also provided by ECNC and WWF. They could be used by the systems to help to the adaptation to the specific domain. For English, there were a total of 113 background documents, containing 2,737,202 words.

We do not apply any kind of specific domain adaptation technique to our supervised class–based system. In order to adapt our supervised system to the environmental domain we just increase automatically the training data with new training examples from the domain. To acquire these examples, we use the 113 background documents of the environmental domain provided by the organizers. We use TreeTagger (Schmid, 1994) to preprocess the documents, performing PoS–tagging and lemmatization. Since the background documents are not semantically annotated, and our supervised system needs labeled data, we have selected only the monosemous instances occurring in the documents according to our BLC20 semantic classes[25]. Note that this approach can only be exploited by class-based WSD systems. In this way, we have obtained automatically a large set of examples annotated with BLC20. This semantic class was selected because it provided very good results in previous experiments. In order to analyze how the same approach and system would work with other level of abstraction, we performed the same evaluation *a posteriori* using BLC50, WordNet Domains and SuperSenses besides to BLC20, which was the official participation in SemEval-2. Nevertheless, this section will be focused on BLC20.

Regarding BLC20, Table 17 presents the total number of training examples extracted from SemCor (SC) and from the background documents (BG). As expected, by this method a large number of monosemous examples can be obtained for nouns and verbs, although, verbs are much less productive than nouns. However, all these background examples correspond to a reduced set of 7,646 monosemous words.

	Nouns	**Verbs**	**N+V**
SC	87,978	48,267	136,245
BG	193,536	10,821	204,357
Total	*281,514*	*59,088*	*340,602*

Table 17: Number of training examples for BLC20

Table 18 lists the ten most frequent monosemous nouns and verbs occurring in the background documents. Remember that all these examples are monosemous according to BLC20 semantic classes.

23. `http://www.ecnc.org`
24. `http://wwf.org`
25. BLC20 (see section 4) stands for Basic Level Concepts obtained with all relations criterion and a minimum threshold of sub–concepts subsumed equal to 20.

	Nouns		Verbs	
	Lemma	**# ex.**	**Lemma**	**# ex.**
1	biodiversity	7,476	monitor	788
2	habitat	7,206	achieve	784
3	specie	7,067	target	484
4	climate	3,539	select	345
5	european	2,818	enable	334
6	ecosystem	2,669	seem	287
7	river	2,420	pine	281
8	grassland	2,303	evaluate	246
9	datum	2,276	explore	200
10	directive	2,197	believe	172

Table 18: Most frequent monosemous words in the background documents

	Nouns	**Verbs**	**N+V**
SC	87,978	48,267	136,245
BG	116,912	7,019	123,931
Total	*204,890*	*55,286*	*260,176*

Table 19: Number of training examples for word senses

Our approach applies the same semantic class architecture shown in the previous sections, but using examples extracted from the background documents. In this case, the semantic class used to extract the examples and generate the classifiers is BLC20[26]. We select a simple feature set widely used in many WSD systems. In particular, we use a window of five tokens around the target word to extract word forms, lemmas; bigrams and trigrams of word forms and lemmas; trigrams of PoS tags, and also the most frequent BLC20 semantic class of the target word in the training corpus.

To analyze the contribution of the monosemous examples in the performance of the system three experiments we have defined:

- BLC20–SC: only training examples extracted from SemCor
- BLC20–BG: only monosemous examples extracted from the background data
- BLC20–SCBG: training examples extracted from SemCor and monosemous background data

The first run (BLC20–SC) aims to show the behavior of a supervised system trained on a general corpus, and tested in a specific domain. The second one (BLC20–BG) analyzes the contribution of the monosemous examples extracted from the background data. Finally, the third run (BLC20-SCBG) studies the robustness of the approach when combining the training examples from SemCor and the automatic ones obtained from the background documents.

Table 20 summarizes ordered by recall the official results of the participants in the English WSD domain specific task of SemEval–2. In this table, Type refers to the approach followed by the corresponding system: Weakly Supervised (WS), Supervised (S) or KB (Knowledge Based, unsupervised). We only participate with the system using BLC20 as semantic class (the BLC20–SC/BG/SCBG runs). The word–based classifiers (labeled Sense–BG, Sense-SC and Sense–SCBG)

26. In this case we use the set of BLCs from WordNet3.0, because also this version of WN was the one used in the annotation.

have been included after the evaluation campaign. Finally, as we mentioned in the introduction, we have also included the performance of ItMakesSense system, which is one of the best performing WSD systems, on this same task for comparison purposes (it is the row on the table called ItMakesSense in Italics).

Rank	System ID	Type	P	R
1	CFILT–2	WS	0.570	0.555
2	CFILT–1	WS	0.554	0.540
3	IIITH1-d.1.ppr.05	WS	0.534	0.528
4	IIITH2-d.2.ppr.05	WS	0.522	0.516
5	**BLC20–SCBG**	**S**	**0.513**	**0.513**
-	*ItMakesSense*	S	*0.510*	*0.510*
6	**BLC20–SC**	**S**	**0.505**	**0.505**
-	*Most Frequent Sense*	-	*0.505*	*0.505*
7	CFILT–3	KB	0.512	0.495
8	Treematch	KB	0.506	0.493
9	Treematch2	KB	0.504	0.491
10	**Sense–SCBG**	**S**	**0.498**	**0.484**
11	**Sense–SC**	**S**	**0.498**	**0.484**
...	...	...	...	...
25	**BLC20–BG**	**S**	**0.380**	**0.380**
...	...	...	...	...
-	*Random baseline*	-	*0.232*	*0.232*
32	**Sense–BG**	**S**	**0.045**	**0.001**

Table 20: Precision and Recall of SemEval–2 participants. ItMakesSense results are included for comparison purpose only

In general, the results reported by SemEval for this task were quite low. The best system only achieved a precision of 0.570, and the most frequent baseline reached a precision of 0.505. This fact shows that the domain adaptation of WSD systems is a very difficult task.

Analyzing the results of our three runs at SemEval, our worst result is obtained by the system using only the monosemous background examples (BLC20–BG). This system ranks 23rd[27] with a Precision and Recall of 0.380 (0.385 for nouns and 0.366 for verbs). The system using only SemCor (BLC20–SC) ranks 6th with Precision and Recall of 0.505 (0.527 for nouns and 0.443 for verbs). This is also the performance of the first sense baseline. As expected, the best result of our three runs is obtained when combining the examples from SemCor and the background (BLC20–SCBG). This supervised system obtains the 5th position with a Precision and Recall of 0.513 (0.534 for nouns, 0.454 for verbs) which is slightly above the baseline. Actually, this version of the system obtains slightly better results than the best performing supervised system (ItMakesSense). Also note that we could include automatically monosemous examples from the background test thanks to the class-based nature of the WSD system.

Moreover, our system is the only one completely supervised participating in the task. The organizers calculated the recall with a confidence interval of 95% using bootstrap re-sampling procedure (Noreen, 1989). This method of estimation might be more strict than other pairwise methods. It reveals that the differences between the four first systems and our system (BLC20–SCBG) are not

27. In the table it appears in the 25th position due to we have included the word–based classifier results.

statistically significant. As can be seen in Figure 9, there is overlapping between the recall confidence interval of the four first systems and our system (ranking the 5th), which proves that the differences are not statistically significant[28].

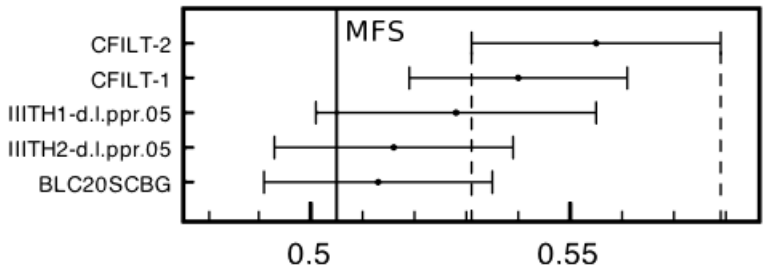

Figure 9: Recall confidence intervals.

Possibly, the reason of low performance of the BCL20–BG system is the high correlation between the features of the target word and its semantic class. In this case, these features correspond to the monosemous word while later are evaluated over polysemous words, with all kind of features. However, it also seems that class-based systems are robust enough to incorporate large sets of monosemous examples from a domain text. In fact, to our knowledge, this is the first time that a supervised WSD algorithm has been successfully adapted to a specific domain. Furthermore, our system trained only on SemCor also achieves a good performance, reaching the most frequent baseline, showing the robustness of class-based WSD approaches to domain variations.

Comparing to word–based classifiers, it seems that our BLC20 classes contribute in two main aspects. First, using the same set of features, the class–based classifiers obtain better results than word–based ones. The classifiers built with BLC20 are more robust and domain adaptable than word–based approaches. Second, the experiment that uses only examples extracted from background data considering word senses (Sense-BG) obtain an accuracy very close to zero, while the same experiment but using BLC20 semantic classes (BLC20–BG) reaches an accuracy of 0.380. This fact indicates that BLCs are useful to extract good training examples from unlabeled data. As mentioned previously, in order to obtain a better insight, after the evaluation campaign we performed the same evaluation with our system using other semantic classes which represent different levels of abstractions: BLC50, WordNet Domains and SuperSenses. Table 21 shows the precision (P) and recall (R)[29] of our evaluation considering different training datasets (SemCor only, Background documents only and both SemCor and Background documents: SC, BG and SC+BG respectively) and different semantic classes.

As can be seen in Table 21, BLC20 leads to a better performance when using the three different corpora for training (BG, SC and SCBG). When training only with monosemous examples extracted from the background documents, BLC20 obtains the best result, which may indicate that its level of abstraction is more adequate than any other, including WND or SS, which are sets much smaller and with a much lower polysemy. The same effect can be drawn from the results when training with SemCor and the monosemous examples from the background (SCBG). the best results are obtained with BLC20, and together with SuperSenses are the only two semantic classes that seem

28. This figure has been taken directly from the overview paper of the task.
29. These figures have been obtained using the official scorer script and the official gold key, without any modification.

System ID	Type	P	R
BLC20–SCBG	S	0.513	0.513
ItMakesSense	S	*0.510*	*0.510*
BLC20–SC	S	0.505	0.505
Most Frequent Sense	S	*0.505*	*0.505*
WND–SC	S	0.495	0.495
Sense–SCBG	S	0.498	0.484
Sense–SC	S	0.498	0.484
SS-SCBG	S	0.484	0.484
BLC50–SCBG	S	0.481	0.481
BLC50–SC	S	0.481	0.481
SS–SC	S	0.472	0.457
WN–SCBG	S	0.471	0.471
BLC20–BG	S	0.380	0.380
WND–BG	S	0.362	0.362
SS–BG	S	0.348	0.348
BLC50–BG	S	0.277	0.277
Random baseline	-	*0.232*	*0.232*

Table 21: Results of our experiments according to different semantic classes

to benefit from the background monosemous examples. These results seem to confirm the potential capabilities of BLC20 to provide an adequate level of abstraction to perform class-based WSD.

Finally, we have proved that our system performs at the same level of one state-of-the-art system[30], the ItMakesSense system (Zhong & Ng, 2010). Considering that the set of features of our system is quite simple, and that we do not apply any machine learning optimization nor feature engineering, our results show that the use of Semantic Classes provides a very robust behavior on specific domains, reaching state-of-the-art results.

10. Concluding Remarks

Word sense disambiguation is a difficult task as empirically has been demonstrated by all SensEval/SemEval exercises. One reason of such difficulties could be the use of inappropriate sets of word meanings. While WordNet is the *de-facto* standard repository of meanings, several attempts have been made grouping its senses in order to achieve higher levels of accuracy. Moreover, this approach tries to ease the hard task of creating large enough sets of annotated data per domain and language to train supervised systems. A possible solution would be to use for manual annotation semantic class labels instead of fine-grained word senses (Schneider, Mohit, Oflazer, & Smith, 2012; Schneider, Mohit, Dyer, Oflazer, & Smith, 2013).

Several attempts have been made to obtain word sense groupings to alleviate the problem of the too fine granularity of word senses, most widely using WordNet senses. In most cases the approach consists in grouping different senses of the same word, resulting in a decrease of the polysemy, while reducing its discriminative capacity. Other works use predefined sets of semantic classes to be integrated directly in a WSD system, mainly SuperSenses.

30. This has been tested offline, as the ItMakesSense system did not participate in the task. We downloaded the last version of the software from `http://www.comp.nus.edu.sg/~nlp/software.html`.

In this work we describe a simple method to automatically select Basic Level Concepts from WordNet. Based on very simple structural properties of WordNet, our method automatically selects different sets of BLC representing different levels of abstraction.

The aim of this work is to explore on several all–words WSD tasks the performance of different levels of abstraction provided by Basic Level Concepts, WordNet Domains, SUMO and SuperSense labels. Furthermore, our study empirically demonstrates that:

a) these word sense groupings cluster senses into a coherent level of abstraction in order to perform supervised class–based WSD while not harming its performance,

b) these semantic classes can be successfully used as semantic features to boost the performance of these classifiers,

c) the class–based approach to WSD reduces dramatically the required amount of training examples to obtain competitive classifiers,

d) the class–based approach obtains competitive performances compared with word-based systems,

e) the class–based approach outperforms word–based systems when evaluated at class level,

f) the robustness of our class-based WSD system when performing out of domain evaluation,

g) our system reaches results comparable to a state-of-the-art system (ItMakesSense) when tested on a specific domain.

In general, class–based disambiguation of nouns and verbs achieves better results than most of the word–based systems presented in both SensEval2 and SensEval3. We also showed that the class-based approach reduces considerably the required amount of training examples. In order to prove that such type of disambiguation is possible and accurate we have ranked the class-based systems together with the SensEval2 and Senseval3 official results. In order to establish a fair comparison we mapped when necessary word senses to semantic classes and *viceversa*.

Some experiments have been designed to use our class–based classifiers to perform word–sense disambiguation. It has been shown that a very simple approach of selecting the first sense in WordNet that corresponds to the class selected by the classifiers performs as well as the top systems at SensEval2 and SensEval3.

Additional experiments have been carried out to compare the word–based systems to perform class–based disambiguation. In this case we translated the official system outputs to its corresponding semantic classes.

Different experiments have been performed using different levels of abstraction, ranging from SuperSenses (a very small set) to SUMO (which has over 1,000 labels linked to WordNet1.6 senses), WordNet Domains (with 163 labels), or Basic Level Concepts (with an arbitrary number of classes depending on the abstraction level selected).

With some expected differences between SensEval2 and SensEval3 results, most of the class–based systems outperform the baselines both for nouns and verbs. Specially for nouns, class-based systems outperforms most of the SensEval2 and SensEval3 systems. In general, the results obtained by SVM-semBLC20 are not very different to the results of SVM-semBLC50. Thus, we can select

a medium level of abstraction, without having a significant decrease of the performance. Considering the number of classes, BLC classifiers obtain high performance rates while maintaining much higher expressiveness than SuperSenses. However, using SuperSenses (40 classes) we can obtain a very accurate semantic tagger with performances around 80%. Even better, we can use BLC20 for tagging nouns (558 semantic classes and F1 over 75%) and SuperSenses for verbs (14 semantic classes and F1 around 75%).

Our systems at SemEval–2 "All-words Word Sense Disambiguation on a Specific Domain" task proved that simple features exploiting BLC can perform as well as more sophisticated methods. Comparing with word–based classifiers, we see that our BLC20 classes contribute in two main aspects: the class–based classifiers obtain better results than word–based ones and semantic classes contribute effectively to those results. This fact indicates that, in particular, BLC20 are useful to extract monosemous training examples from unlabeled domain data.

Our next goal is to exploit the inconsistencies of the different labeling provided by the different class-based classifiers in order to obtain a more robust and accurate class-based WSD system. The main idea is to study why several classifiers, each one based on a different degree of abstraction (e.g. BLC20, BLC50, WordNet Domains, etc.) label a concrete context or example with incompatible tags. In this manner, we would be able to predict when to apply the best classifier depending on the context.

Acknowledgements

This work has been partially supported by the NewsReader project[31] (ICT-2011-316404), the Spanish project SKaTer[32] (TIN2012-38584-C06-02).

References

Agirre, E., & de Lacalle, O. L. (2003). Clustering wordnet word senses. In *Proceedings of RANLP'03*, Borovets, Bulgaria.

Agirre, E., & Edmonds, P. (2007). *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Agirre, E., López de Lacalle, O., Fellbaum, C., Hsieh, S.-K., Tesconi, M., Monachini, M., Vossen, P., & Segers, R. (2010). Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 75–80, Uppsala, Sweden. Association for Computational Linguistics.

Bhagwani, S., Satapathy, S., & Karnick, H. (2013). Merging word senses. In *Proceedings of Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-8)*, pp. 11–19.

Castillo, M., Real, F., & Rigau, G. (2004). Automatic assignment of domain labels to wordnet. In *Proceeding of the 2nd International WordNet Conference*, pp. 75–82.

Ciaramita, M., & Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, pp. 594–602, Sydney, Australia. ACL.

31. http://www.newsreader-project.eu
32. http://nlp.lsi.upc.edu/skater

Ciaramita, M., & Johnson, M. (2003). Supersense tagging of unknown nouns in wordnet. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP'03)*, pp. 168–175. ACL.

Curran, J. (2005). Supersense tagging of unknown nouns using semantic similarity. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pp. 26–33. ACL.

Escudero, G., Màrquez, L., & Rigau., G. (2000). An Empirical Study of the Domain Dependence of Supervised Word Sense Disambiguation Systems. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC*, Hong Kong, China.

Fellbaum, C. (Ed.). (1998). *WordNet. An Electronic Lexical Database*. The MIT Press.

Gangemi, A., Nuzzolese, A. G., Presutti, V., Draicchio, F., Musetti, A., & Ciancarini, P. (2012). Automatic typing of dbpedia entities. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part I*, ISWC'12, pp. 65–81, Berlin, Heidelberg. Springer-Verlag.

González, A., Rigau, G., & Castillo, M. (2012). A graph-based method to improve wordnet domains. In *Computational Linguistics and Intelligent Text Processing*, pp. 17–28. Springer.

Hamp, B., Feldweg, H., et al. (1997). Germanet-a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pp. 9–15. Citeseer.

Hearst, M., & Schütze, H. (1993). Customizing a lexicon to better suit a computational task. In *Proceedingns of the ACL SIGLEX Workshop on Lexical Acquisition*, Stuttgart, Germany.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes: The 90 In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pp. 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Izquierdo, R., Suarez, A., & Rigau, G. (2007). Exploring the automatic selection of basic level concepts. In et al., G. A. (Ed.), *International Conference Recent Advances in Natural Language Processing*, pp. 298–302, Borovets, Bulgaria.

Izquierdo, R., Suárez, A., & Rigau, G. (2009). An empirical study on class-based word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pp. 389–397, Stroudsburg, PA, USA. Association for Computational Linguistics.

Izquierdo, R., Suárez, A., & Rigau, G. (2010). Gplsi-ixa: Using semantic classes to acquire monosemous training examples from domain texts. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 402–406. Association for Computational Linguistics.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., & Rouveirol, C. (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, No. 1398, pp. 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE.

L. Bentivogli, P. Forner, B. M., & Pianta, E. (2004). Revising wordnet domains hierarchy: Semantics, coverage, and balancing. In *COLING 2004 Workshop on Multilingual Linguistic Resources*, Geneva, Switzerland.

Magnini, B., & Cavaglià, G. (2000). Integrating subject field codes into wordnet. In *Proceedings of LREC*, Athens. Greece.

Màrquez, L., Escudero, G., Martínez, D., & Rigau, G. (2006). Supervised corpus-based methods for wsd. In *E. Agirre and P. Edmonds (Eds.) Word Sense Disambiguation: Algorithms and applications.*, Vol. 33 of *Text, Speech and Language Technology*. Springer.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. In *In 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain.*

Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. In *Proceedings of NAACL HLT 2007*.

Mihalcea, R., Csomai, A., & Ciaramita, M. (2007). Unt-yahoo: Supersenselearner: Combining senselearner with supersense and other coarse semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pp. 406–409, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihalcea, R., & Moldovan, D. (2001). Automatic generation of coarse grained wordnet. In *Proceding of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, USA.

Miller, G., Leacock, C., Tengi, R., & Bunker, R. (1993). A Semantic Concordance. In *Proceedings of the ARPA Workshop on Human Language Technology*.

Navigli, R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 105–112, Morristown, NJ, USA. Association for Computational Linguistics.

Navigli, R. (2009). Word Sense Disambiguation: a survey. *ACM Computing Surveys*, *41*(2), 1–69.

Navigli, R., Litkowski, K., & Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 30–35, Prague, Czech Republic. Association for Computational Linguistics.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pp. 17–19. Chris Welty and Barry Smith, eds.

Niles, I., & Pease, A. (2003). Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In Arabnia, H. R. (Ed.), *Proc. of the IEEE Int. Conf. on Inf. and Knowledge Engin. (IKE 2003)*, Vol. 2, pp. 412–416. CSREA Press.

Noreen, E. (1989). *Computer-intensive methods for testing hypotheses: an introduction*. A Wiley Interscience publication. Wiley.

Paaß, G., & Reichartz, F. (2009a). Exploiting semantic constraints for estimating supersenses with crfs.. In *SDM*, pp. 485–496. SIAM.

Paaß, G., & Reichartz, F. (2009b). Exploiting semantic constraints for estimating supersenses with crfs.. In *SDM*, pp. 485–496. SIAM.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., & Dang, H. T. (2001). English tasks: All-words and verb lexical sample. In *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL'2001/EACL'2001*, Toulouse, France.

Peters, W., Peters, I., & Vossen, P. (1998). Automatic sense clustering in eurowordnet. In *First International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain.

Picca, D., Gliozzo, A. M., & Ciaramita, M. (2008). Supersense tagger for italian.. In *LREC*. Citeseer.

Pradhan, S., Dligach, E. L. D., & Palmer, M. (2007). Semeval-2007 task 17: English lexical sample, srl and all words. In *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 87–92, Morristown, NJ, USA. Association for Computational Linguistics.

Rosch, E. (1977). Human categorisation. *Studies in Cross-Cultural Psychology*, *I*(1), 1–49.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pp. 44–49.

Schneider, N., Mohit, B., Dyer, C., Oflazer, K., & Smith, N. A. (2013). Supersense tagging for arabic: the mt-in-the-middle attack.. In *HLT-NAACL*, pp. 661–667. Citeseer.

Schneider, N., Mohit, B., Oflazer, K., & Smith, N. A. (2012). Coarse lexical semantic annotation with supersenses: an arabic case study. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 253–258. Association for Computational Linguistics.

Segond, F., Schiller, A., Greffenstette, G., & Chanod, J. (1997). An experiment in semantic tagging using hidden markov model tagging. In *ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pp. 78–81. ACL, New Brunswick, New Jersey.

Snow, R., S., P., D., J., & A., N. (2007). Learning to merge word senses. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1005–1014.

Snyder, B., & Palmer, M. (2004). The english all-words task. In Mihalcea, R., & Edmonds, P. (Eds.), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43, Barcelona, Spain. Association for Computational Linguistics.

Tsvetkov, Y., Schneider, N., Hovy, D., Bhatia, A., Faruqui, M., & Dyer, C. (2014). Augmenting english adjective senses with supersenses. In *Proc. of LREC*, pp. 4359–4365.

Villarejo, L., Màrquez, L., & Rigau, G. (2005). Exploring the construction of semantic class classifiers for wsd. In *Proceedings of the 21th Annual Meeting of Sociedad Espaola para el Procesamiento del Lenguaje Natural SEPLN'05*, pp. 195–202, Granada, Spain. ISSN 1136-5948.

Vossen, P. (Ed.). (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* . Kluwer Academic Publishers .

Wikipedia (2015). Wikipedia, the free encyclopedia. `https://en.wikipedia.org`.. [Online; accessed 21-August-2015].

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*.

Zhong, Z., & Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pp. 78–83, Stroudsburg, PA, USA. Association for Computational Linguistics.